

PROPOSED CLAIM AMENDMENTS

Claim 1. A software processing method comprising:

a monitoring step, which monitors a status of use of a resource identified as used by a process for a processor;

a process identifying step, in which a compiler inserts a ~~program~~ command that identifies a process using a resource into an input program;

a determining step, in which the compiler inserts into the input program a ~~program~~ command that determines a status of use of the identified resource based upon contention information obtained in the monitoring step; and

a substituting step, in which the compiler inserts a ~~program~~ command that substitutes an equivalent process for the process identified by the ~~program~~ command inserted in the process identifying step, based upon results of the ~~program~~ command inserted into the input program in the determining step; wherein

the equivalent process is equivalent to the process identified by the ~~program~~ command inserted in the process identifying step, and makes reduced use of the resource.

[NOTE: the Applicants agree with most of the proposed changes to claim 1. However, they do not agree with amending the last paragraph to recite “reduced use of the status of the resource,” as the meaning of the proposed phrase is not understandable]

Claim 16. A software processing method comprising:

a monitoring step for status of use, which monitors the status of use of a second processor, the second processor performing processing in response to a processing request by a first processor; [[and]]

an identifying step, in which a compiler identifies a process containing command to use the second processor;

a mapping step, in which the compiler maps a first process providing the processing request of the first processor to the second processor into a first memory bank and a second process executed by the first processor into the first memory bank and a second process executed by the first processor into a second memory bank; and

an altering step for software processes, which alters software processing processes executed by the first processor or the second processor in response to contention information, the contention information being obtained in the monitoring step for status of use, wherein

the first processor can access a plurality of memory banks of a memory only for the first processor by using a same address, and the plurality of memory banks includes [[a]] the first memory bank and [[a]] the second memory bank, ~~the first memory bank including a program providing the processing request of the first processor to the second processor, and the second memory bank including a program executed by the first processor; and~~

the contention information is a signal indicating a switch from the first memory bank to the second memory bank.

[NOTE: the above amendments to claim 16 are based on the comments made by telephone, noting claim 1, FIG. 35, and paragraph [0216] of the PGPUB for this application. As suggested, an "identifying" step is added. However, the proposal to add a "resource" to the claim is unnecessary in view of the recited "second processor."]

Claim 18. A software processing method comprising:

a monitoring step, which monitors a status of use of a resource identified as used by a process for a processor;

a process identifying step, in which a compiler inserts a ~~program~~ command that identifies a process using a resource into an input program;

a storing step for storing contention information obtained in the monitoring step at a current time;

a determining step, in which the compiler inserts into the input program a ~~program~~ command that determines a status of use of the identified resource based upon contention information stored at a past time; and

a substituting step, in which the compiler inserts a ~~program~~ command that substitutes an equivalent process for the process identified by the ~~program~~ command inserted in the process identifying step, based upon results of the ~~program~~ command inserted into the input program in the determining step; wherein

10/727,055

the equivalent process is equivalent to the process identified by the ~~program~~ command inserted in the process identifying step, and makes reduced use of the resource.

[NOTE: claim 18 is amended in a manner similar to claim 1]